determined. An aspect ratio of the zoom portion defined by the user input is adjusted to correspond to the zoom display device resolution. The display controller system is programmed to implement the display surface zoom to provide a full screen view of the zoom portion on the zoom display device. In the display controller system, the zoom portion of the frame buffer memory is scaled, converted into a display signal and output.

BB

IN THE DRAWINGS

Proposed changes to Figs. 1, 2, 3, 4, 5, 6, 7 and 8 are submitted herewith for the Examiner's approval.

REMARKS

Addressing the issues raised by the Examiner in his Action in the same order, Applicants respond as follows.

The abstract has been replaced by a new text in compliance with MPEP§608.01(b).

The phrase "portion respect" at page 4, line 2 has been amended as suggested by the Examiner.

The reference to Figure 3 at page 7, line 28 has been amended to refer to Figure 2.

A description of Figure 3 was provided in the originally filed specification in the paragraph beginning at page 9, line 10, however, this paragraph has been amended to refer correctly to Figure 3 (not 4), and to include the sentence describing the operation found in Figure 3 as filed with this application. The proposed amendment to Figure 3 deletes the corresponding text in the drawings.

The paragraph at page 4, line 10, has been amended to include textual support from claims 3,4,23 and 24.

The reference numeral 14 for the backend scalers has been deleted from the specification.

Reference numeral 60 for the 3D drawing engine has been added to proposed amended Figure 1.

The reference numerals 21,22,23,24,31,32,70 and 71 have been removed from Figures 3 to 8.

The Examiner has indicated that Figure 6 was not provided by Applicant. A marked-up copy of Applicants' original Figure 6 is submitted herewith. As the Examiner will appreciate, the drawing submitted herewith is essentially the same as Figure 5 with the exception that a third zoom buffer is shown within memory 50, along with a copy of the zoomed hardware cursor in accordance with the reference to Figure 6 at page 10, line 26. While Applicants' source drawing refers to the third zoom buffer as a "Zoom Front Buffer", Applicants' are removing the word "Front" from the Figure 6 to avoid any appearance of adding unsupported subject matter.

Reference numerals have been added to identify the steps shown in the flow chart of Figure 2, and references to these steps and, where suitable, phrase or sentences corresponding to the text of Figure have been added to the amended specification.

A period has been added to claim 16.

Claims 17 and 18 have been amended to remove the word "fractional".

Prior to addressing the other claim amendments, Applicants wish to clarify an additional point concerning the specification and claims. During the interview, Applicants' below-signed agent discussed with the Examiner a proposal to change "cursor" to "pointer" in the present specification and claims. Further consideration of the need to clarify this term in the present application has shown that the term "cursor" should be clearly understood by a person skilled in the art without confusion. The IBM Dictionary of Computing, tenth edition (August 1994), at page 159, provides as definition number 5 for the entry "cursor": "A primitive, such as an arrowhead, that can be moved about the screen by means of an input device, typically a mouse."

Claim 1 has been amended to more clearly define Applicants' invention. More specifically, the claims have been amended so that there would be no ambiguity that Applicant's invention involves the use of a display controller system to perform a zoom on a user selected portion of a surface in the display controller system memory, namely the frame buffer memory. The objective is to provide full screen zoom output of the user selected portion.

The prior art cited by the Examiner was reviewed during the Interview. Generally, the prior art fails to teach applying a zoom to a user selected portion of a surface in which the zoom is performed on the frame buffer memory surface using display controller system resources to provide a full screen output. The combination of user zoom control with the programming the display controller system to provide full screen output of a portion of the frame buffer memory using the controller system's scaling abilities is not taught or suggested by the references of record.

Independent claim 21 defines an invention incorporating the same inventive features as claim 1. In the case of claim 21, however, the user-defined portion is a non-integer factional portion of the frame buffer memory, meaning that scaling will not be achieved by changing screen resolution alone, and the use of a scaler will be involved to interpolate pixel values. The zoom portion could also be defined as a panning window that follows the mouse.

Support for new claims 28 through 31 was discussed with the Examiner in the Interview. Claims 28 and 31 as presented in this amendment follow more closely the specification at page 8, line 10, and are clearly supported by the specification.

In view of the foregoing, a Notice of Allowance for claims 1 to 33 is respectfully requested.

Respectfully submitted,
Kamran AHMED et al.

By:

James Anglehart (Reg. 38,796)

(514) 847-4244

Marked-up copy of specification

Page 3, line 28

There are two basic ways of determining the resolution of the zoom display device and adjusting the aspect ratio of the portion. The first way is to determine the suitable aspect ratio based on the resolution of the zoom display and to force the user selection of the frame portion to choose a frame portion of the same aspect ratio. The second way is to allow the user to define any frame portion and then to adjust the frame portion with respect to the aspect ratio based on the resolution of the zoom display.

Page 4, line 10

The zoom on the second display can be filtered to avoid pixelation (i.e. an image not illustrating coarse pixels) or unfiltered (i.e. an image illustrating coarse pixels) to facilitate pixel by pixel viewing and editing. User input can also define a user's choice of filtering or non-filtering.

Page 6, line 24

Figure 1 shows a high level block diagram of the preferred embodiment. Two CRTC's 11 and 12 are capable of fetching one or more display surfaces from a single frame buffer memory (50) which can be SGRAM, SDRAM, or any other type of Random Access Memory (RAM). Each CRTC may also contain one or more backend scalers ~~14 (refer to Figure 2)~~ that allows the input surfaces to be re-scaled. While, within the context of the present invention, each controller 11 and 12 does not need to access more than one surface, greater image processing and display ability may be provided when multiple surfaces can be accessed by each controller.

Page 7, line 28

Figure 2~~3~~ shows a flow chart for the embodiment illustrated in Figs. 4 and 5. When the end user enables the zoom using a hotkey (H1) or the like in step 100, the software allows the user to select a rectangular window from the primary display in step 101. Once the zoom operation is enabled, the hotkey is also detected (step 100') to determine if the zoom operation should be disabled (step 113). One example of this could be that the user holds down the mouse key at which point the coordinates of one corner of the zoom window are determined. The user then drags the mouse while holding down the key and stops at the corner diagonally opposite the first one to specify the rectangle and lets go of the key. At this point the coordinates of the corner diagonally opposite the first one are determined and this information is enough to specify the size and location of the zoom window. Of course, there are many other ways to determine this rectangular area without departing from the spirit and scope of the invention. The coordinates of the zoom window (including address in memory) are thus stored. As illustrated in Fig. 2, the coordinates of the zoom window are sent to the display driver in step 102.

Page 8, line 10

The resolution of the destination can be either automatically calculated or user defined (step 103). When it is user defined, the software uses this resolution (step 104). In the preferred embodiment, it is automatic, it could be chosen in a variety of ways ranging from (but not limited to) the closest standard resolution (to the resolution of the zoom window) to the largest resolution possible etc. These resolutions determination options can also be specified by the user. Once the destination resolution is chosen, the scaling factor is determined (step 106). This determination of the scaling factor is within the general knowledge of those in skilled in the art.

Page 9, line 10

With knowledge of the destination resolution, a buffer of this resolution is reserved in memory (step 105) for the zoomed area (zoom buffer). As will be appreciated, multiple buffers can be allocated if double or triple buffering is desired and when multiple zoom windows are defined. The secondary CRTC is then programmed (step 107) to read from this zoom buffer (or set of zoom buffers). If the panning or ~~(~~mouse following~~)~~ feature is enabled (step 108) then the location of the zoom window is consistently updated (step 109), see Figure 3~~4~~. In this embodiment, the zoom window can be locked to the movement of the mouse and the zoomed area is updated in real-time.

Page 11, line 12

Once the use of filtering or no-filtering is decided (step 110), the 3D drawing engine is used to provide filtering (step 111), or the 3D drawing engine is used to provide scaling without filtering. Alternatively to using the 3D drawing engine 60, the backend scaler ~~14~~ of CRTC2 12 can also be used to scale the zoomed window (see Figure 8). The CRTC2 12 is set to read from the location where the zoom window is located and the scaler is programmed to scale using the determined scale factor. The zoom window can be fetched directly from the main display buffer or the zoom window can be copied (blit) into another region in memory and the CRTC2 (12) can read from there (see Figure 8). In this case, the control of filtering and non-filtering, will depend on the filtering capabilities of the specific scaling unit used.

Marked-up copy of abstract

~~A unique combination of hardware and software enhancement building on a generic single chip multi-display graphics subsystem. Multiple independent displays from one graphics controller can be driven in a wide variety of modes using multiple display controllers. Digital content creation, desktop publishing and web browsing amongst other applications require the user to view or edit display images (or data etc) of varying detail and formats (text or image for example). Often the user is more interested in viewing or editing specific areas of the display. As an example, a user editing a photograph using a photo-editing software might want to zoom into an area enough to edit individual pixels of the image. The user has no way of seeing how the edits on the zoomed area affect the entire image without having to toggle back and forth between the zoomed and un-zoomed image. Some software allows the possibility of showing the entire image in a little window in the corner. However, the small size of such window rarely makes up for the inconvenience.~~ A display controller system is controlled to provide a display surface zoom using hardware scaling from user input at the operating system, application program or hardware level. User input defining coordinates of a frame portion within a frame buffer memory is obtained, and a resolution of the zoom display device is determined. An aspect ratio of the zoom portion defined by the user input is adjusted to correspond to the zoom display device resolution. The display controller system is programmed to implement the display surface zoom to provide a full screen view of the zoom portion on the zoom display device. In the display controller system, the zoom portion of the frame buffer memory is scaled, converted into a display signal and output.

Marked up copy of claims in accordance with 37CFR§1.121(c)(ii)

1. (twice amended) A method of controlling a display controller system to provide a display surface zoom, said display controller system having a ~~main surface~~frame buffer memory and output to at least one zoom display device, the method comprising the steps of:

receiving user input defining coordinates of a fixed position frame portion within said ~~main surface~~frame buffer memory;

determining a resolution of said at least one zoom display device and adjusting an aspect ratio of said portion defined by said user input to correspond to said resolution;

programming said display controller system to implement said display surface zoom to provide a full screen view of said portion on said at least one zoom display device;

in said display controller system, scaling said portion of said ~~main surface~~frame buffer memory~~ in said display controller system~~;

in said display controller system, converting said scaled portion of said ~~main surface~~frame buffer memory into a display signal~~ in said display controller system~~; and

outputting said display signal from said display controller system to said at least one zoom display device.

2. (unamended) The method as claimed in claim 1, wherein said step of converting includes incorporating a representation of a cursor in said display signal, said cursor having a position defined by a cursor position memory used for said ~~main surface~~frame buffer memory.

3. (unamended) The method as claimed in claim 1, further comprising a step of filtering said portion to provide for an image not illustrating coarse pixels.

4. (unamended) The method as claimed in claim 3, wherein said user input further defines a user's choice of filtering or non-filtering.

5. (unamended) The method as claimed in claim 1, wherein said user input further includes a cursor control device input used to control a cursor, and said portion is caused to be dragged or moved over said ~~main surface~~frame buffer memory by movement of said cursor.

6. (unamended) The method as claimed in claim 1, wherein said scaling comprises using a drawing engine associated with said display controller system to scale said portion into a buffer.

7. (unamended) The method as claimed in claim 1, wherein said scaling comprises using a backend scaler associated with said display controller system to scale said portion.

8. (unamended) The method as claimed in claim 7, wherein said scaling further comprises using a backend scaler associated with said display controller system to scale a hardware cursor associated with said portion.

9. (unamended) The method as claimed in claim 6, wherein said scaling further comprises using a drawing engine associated with said display controller system to scale a hardware cursor associated with said portion into a separate hardware cursor buffer.

10. (unamended) The method as claimed in claim 6, wherein said scaling further comprises using a drawing engine associated with said display controller system to scale a hardware cursor associated with said portion and overlay it onto said buffer.

11. (unamended) The method as claimed in claim 6, wherein said image data is stored alternatingly in one of a plurality of buffers, said step of converting comprising reading said image data alternatingly from one of said buffers so as to reduce image flicker and ensure complete buffer update before displaying.

12. (amended) The method as claimed in claim 1, wherein said display controller system comprises a single display output, and said user input causes said single display to switch between displaying said portion and displaying essentially all of said ~~main surface~~frame buffer memory, whereby said zoom is provided independently of an application program.

13. (amended) The method as claimed in claim 1, wherein said display controller system comprises at least two displays outputs, a first one of which displaying essentially all of said ~~main surface~~frame buffer memory, and a second one of which displaying said scaled portion in a full screen view.

14. (unamended) The method as claimed in claim 13, wherein said second display has a different image resolution than an image resolution of said first display, said converting comprising automatically adjusting an image resolution of said signal representing said portion to match said image resolution of said second display.

15. (unamended) The method as claimed in claim 1, wherein said step of receiving user input comprises:

receiving input defining at least two portions of said main display surface to be selectively displayed on one of said at least one zoom display device; and

receiving input selecting one of said at least two portions of said main display surface to be displayed on said one of said at least one zoom display device.

16. (amended) The method as claimed in claim 15, wherein said user input causes a toggling between said portions.

17. (amended) The method as claimed in claim 15, wherein said step of receiving user input further comprises:

associating said input defining said at least one said portion with one of a plurality of application programs,

wherein said step of receiving input selecting one of said at least two ~~fractional~~ portions comprises determining which one of a plurality of application programs is currently active and providing output to said ~~main surface~~frame buffer memory in order to select from at least one of said portions of said main display surface associated with ~~the~~ said currently active one of said plurality of said application programs ~~currently outputting to said main display surface~~.

18. (amended) The method as claimed in claim 17, wherein a change in application program currently active and outputting to said main display surface is detected and caused to automatically change selection of said at least one of said at least two ~~fractional~~ portions.

19. (unamended) The method as claimed in claim 1, wherein said step of receiving user input comprises:

receiving input defining a plurality of portions of said main display surface to be selectively displayed on different zoom display devices; and

receiving input selecting one of said portions of said main display surface to be displayed on each one of said zoom display devices.

20. (unamended) The method as claimed in claim 19, wherein said user input causes a toggling between said portions.

21. (twice amended) A method of controlling a display controller system to provide a display surface zoom, said display controller system having a ~~main surface~~frame buffer memory and output to at least one zoom display device, the method comprising the steps of:

receiving user input defining coordinates of a fractional portion of said ~~main surface~~frame buffer memory to be scaled and displayed, said fractional portion being a non-integer fraction of said ~~main surface~~frame buffer memory;

determining a resolution of said at least one zoom display device and adjusting an aspect ratio of said portion defined by said user input to correspond to said resolution;

programming said display controller system to implement said display surface zoom to provide full screen view of said portion on said at least one zoom display device;

scaling said portion of said ~~main surface~~frame buffer memory;

converting said scaled portion of said ~~main surface~~frame buffer memory into a display signal; and

outputting said display signal to said at least one zoom display device.

22. (unamended) The method as claimed in claim 21, wherein said step of converting includes incorporating a representation of a cursor in said display signal, said cursor having a position defined by a cursor position memory used for said ~~main surface~~frame buffer memory.

23. (unamended) The method as claimed in claim 21, further comprising filtering said portion to provide for an image not illustrating coarse pixels.

24. (unamended) The method as claimed in claim 23, wherein said user input further defines a user's choice of filtering or non-filtering.

25. (unamended) The method as claimed in claim 21, wherein said user input further includes a pointing device output used to control a cursor, and said portion is caused to be dragged or moved over said ~~main surface~~frame buffer memory by movement of said cursor.

26. (unamended) The method as claimed in claim 21, wherein said scaling comprises using a drawing engine associated with said display controller system to generate image data corresponding to said portion.

27. (unamended) The method as claimed in claim 21, further comprising a step of accepting user input adjusting said non-integer fraction to be increased and to be decreased, wherein said user input can cause a zoom magnification to vary upwards and downwards.

28. (new) The method as claimed in claim 1, wherein the step of determining the resolution of the at least one display device comprises automatically a standard resolution of the at least one display device being closest to a resolution of said portion, said step of programming including specifying to said display controller system said closest standard resolution.

29. (new) The method as claimed in claim 28, wherein said display controller system has full-screen output to a main display device and to said at least one zoom display device.

30. (new) The method as claimed in claim 28, wherein said at least one zoom display device comprises a CRT display.

31. (new) The method as claimed in claim 21, wherein the step of determining the resolution of the at least one display device comprises automatically choosing a standard resolution of said at least one display device being closest to a resolution of said portion, said step of programming including specifying to said display controller system said closest standard resolution.

32. (new) The method as claimed in claim 31, wherein said display controller system has full-screen output to a main display device and to said at least one zoom display device.

33. (new) The method as claimed in claim 31, wherein said at least one zoom display device comprises a CRT display.

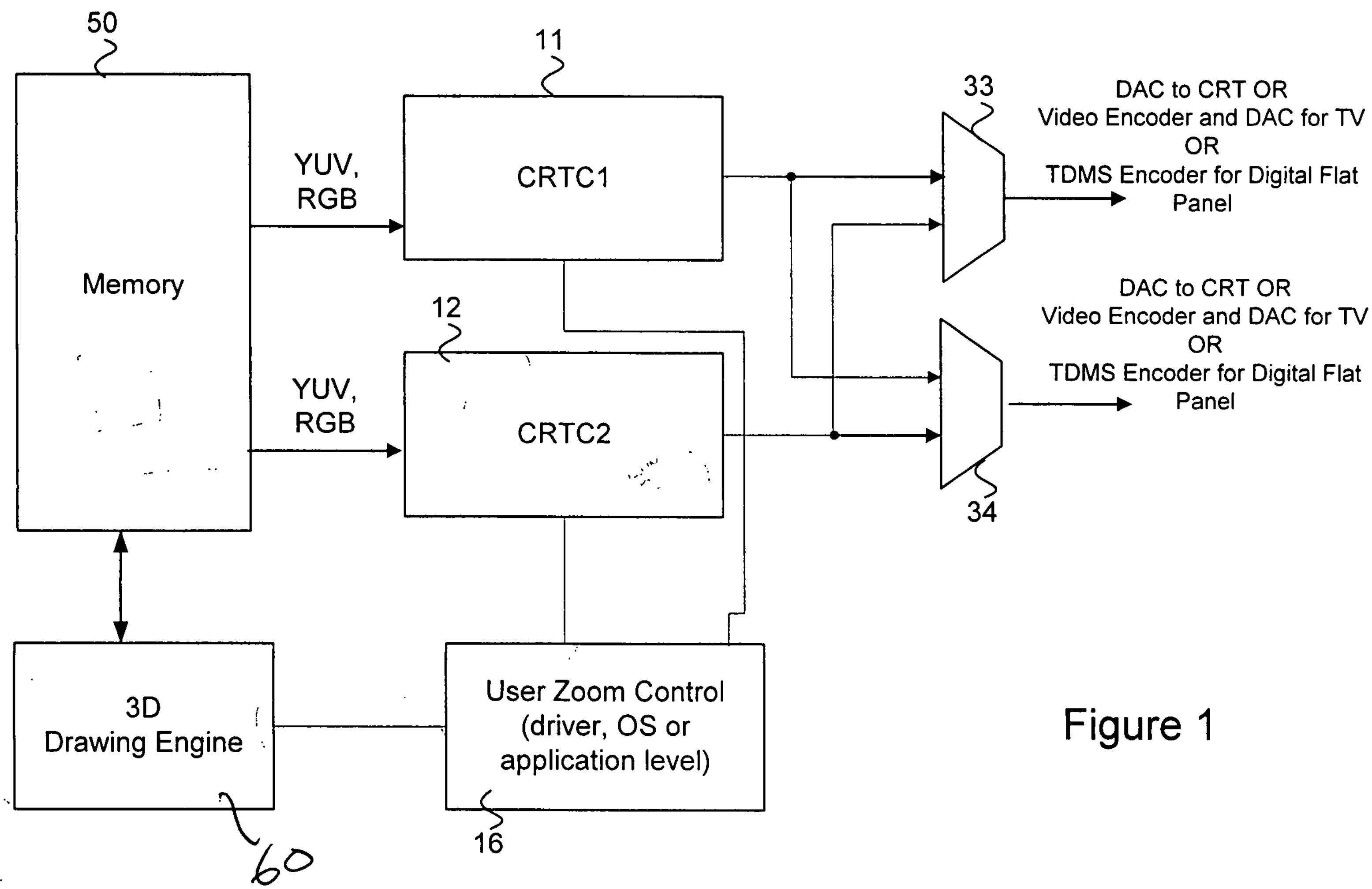
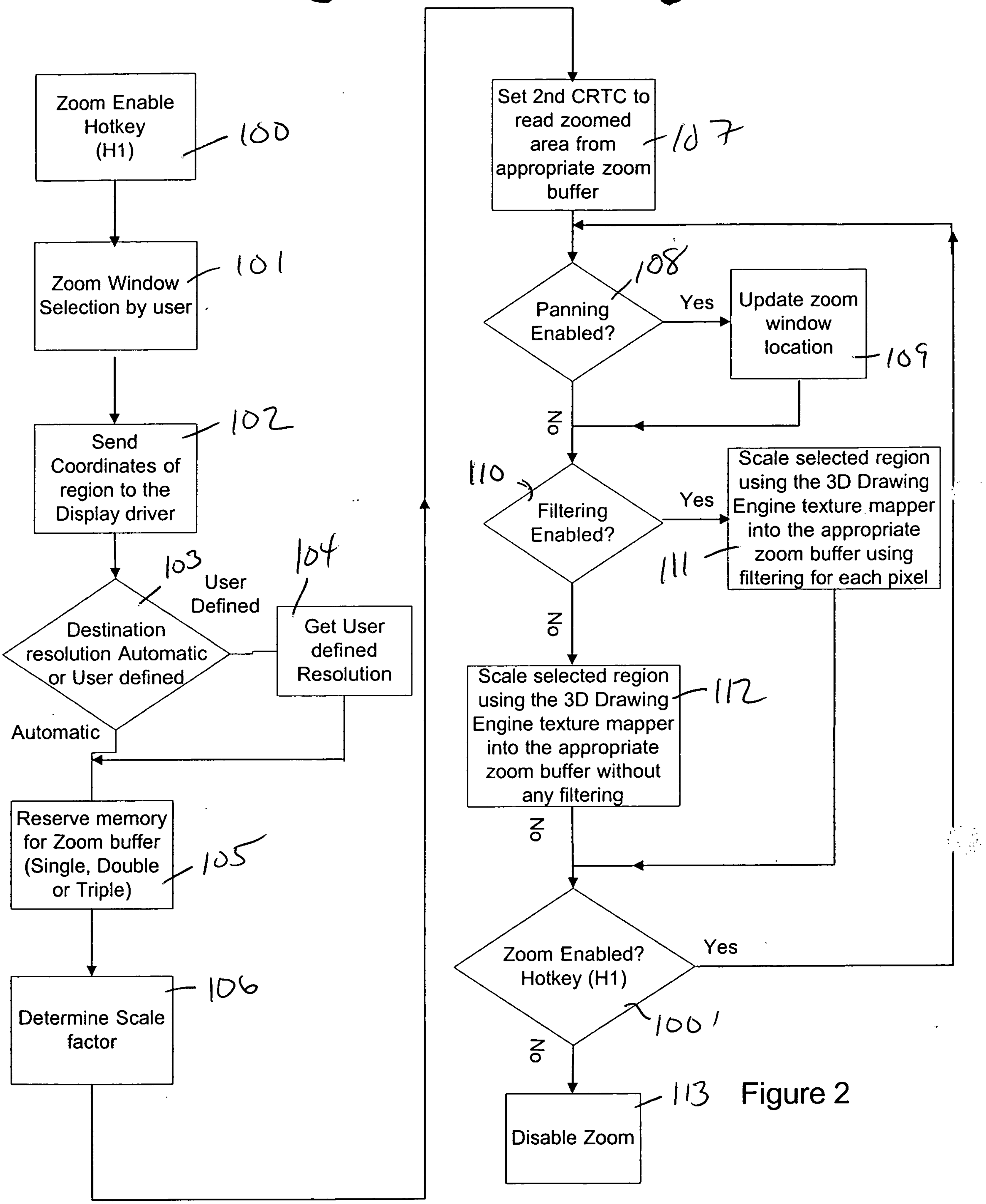

50
11
33
DAC to CRT OR
Video Encoder and DAC for TV
OR
TDMS Encoder for Digital Flat
Panel
Memory
YUV,
RGB
CRTC1
12
DAC to CRT OR
Video Encoder and DAC for TV
OR
TDMS Encoder for Digital Flat
Panel
YUV,
RGB
CRTC2
34
3D
Drawing Engine
User Zoom Control
(driver, OS or
application level)
16
60

Figure 1



Zoom Enable Hotkey (H1)
100
Zoom Window Selection by user
101
Send Coordinates of region to the Display driver
102
103
Destination resolution Automatic or User defined
User Defined
104
Get User defined Resolution
Automatic
Reserve memory for Zoom buffer (Single, Double or Triple)
105
Determine Scale factor
106
Set 2nd CRTC to read zoomed area from appropriate zoom buffer
107
108
Panning Enabled?
Yes
Update zoom window location
109
No
110
Filtering Enabled?
Yes
Scale selected region using the 3D Drawing Engine texture mapper into the appropriate zoom buffer using filtering for each pixel
111
No
Scale selected region using the 3D Drawing Engine texture mapper into the appropriate zoom buffer without any filtering
112
No
Zoom Enabled? Hotkey (H1)
Yes
100'
No
Disable Zoom
113

Figure 2

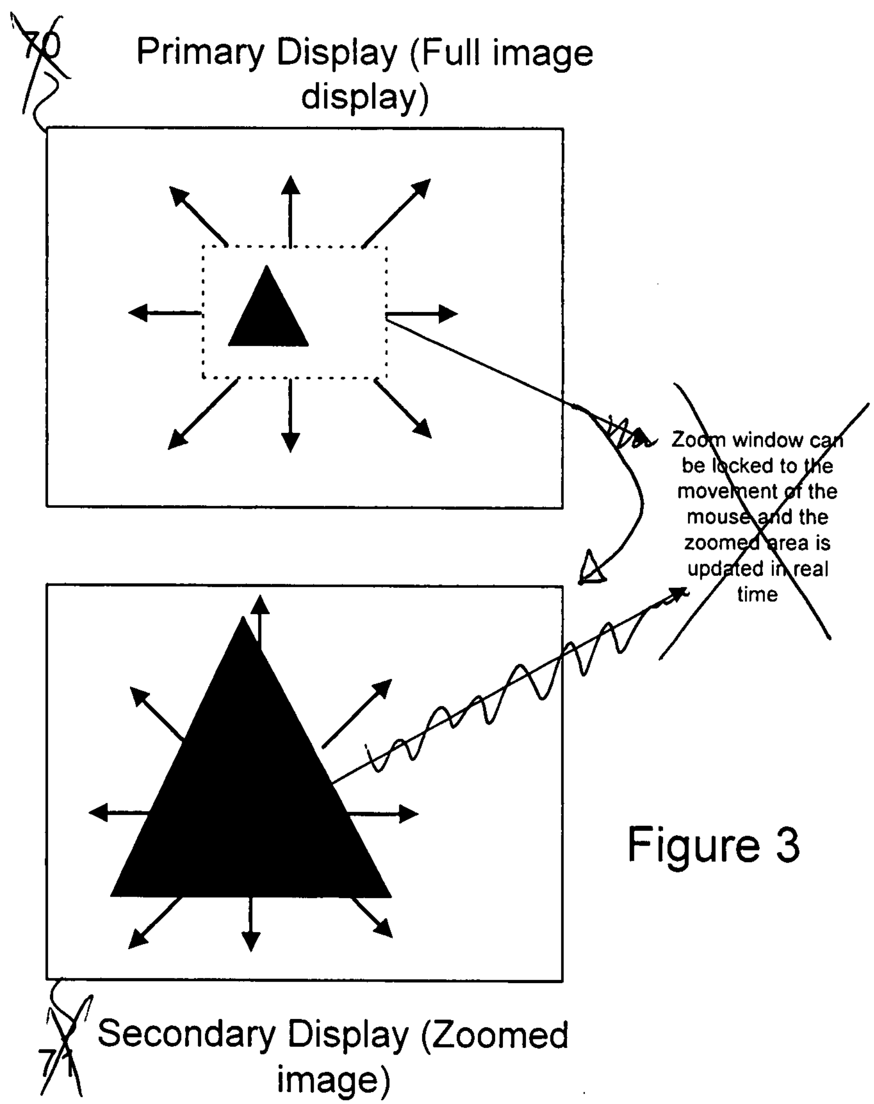

70
Primary Display (Full image display)
Zoom window can be locked to the movement of the mouse and the zoomed area is updated in real time
71
Secondary Display (Zoomed image)

Figure 3

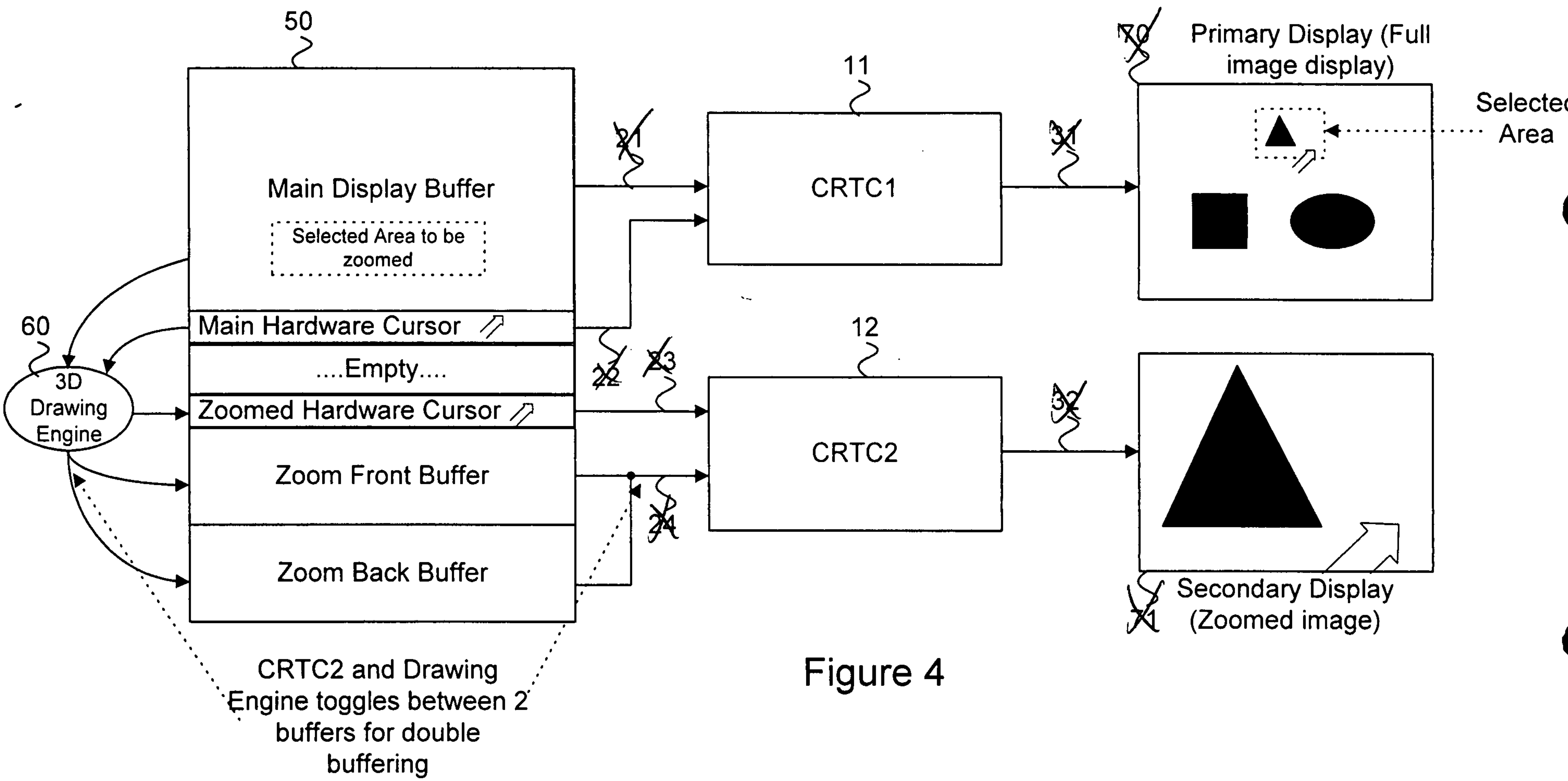
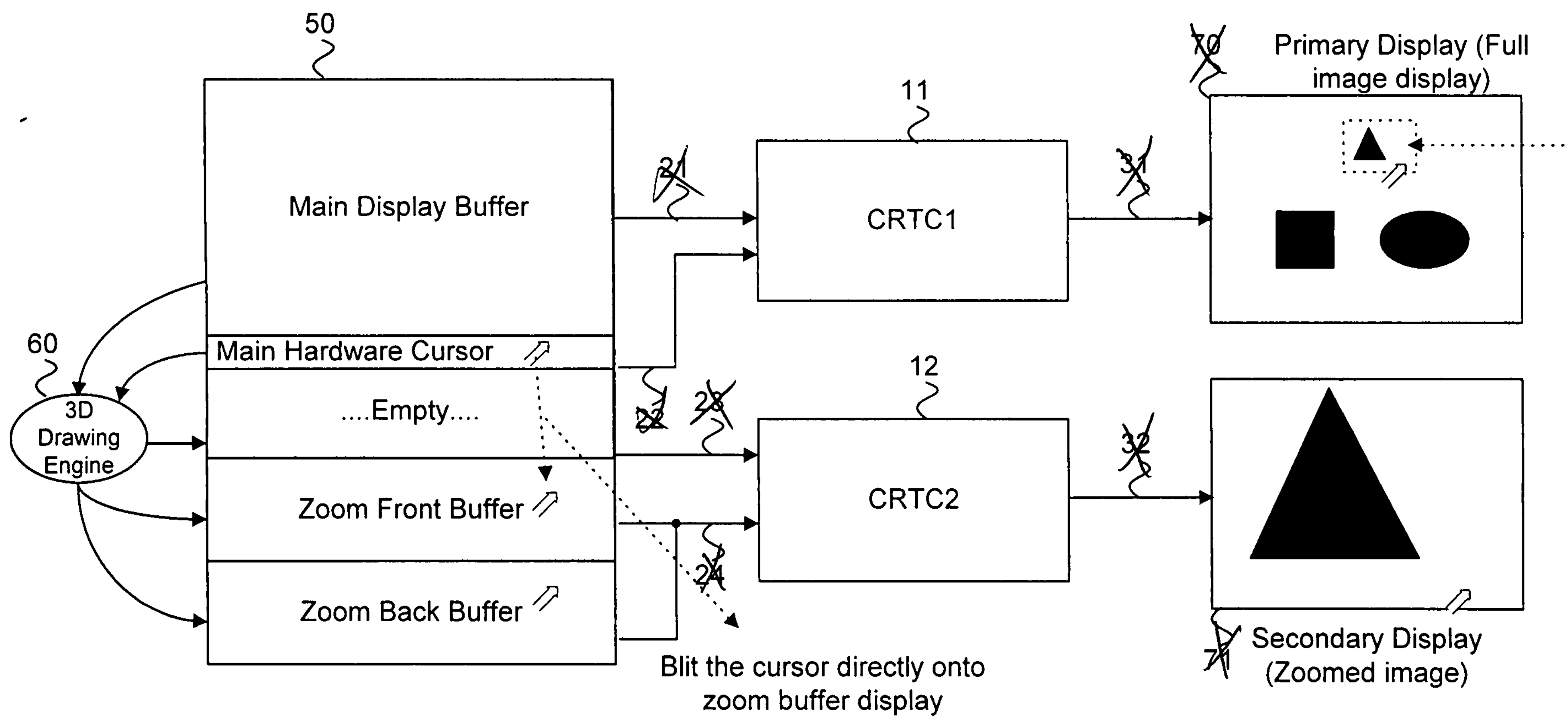

50
Main Display Buffer
Selected Area to be zoomed
Main Hardware Cursor
....Empty....
Zoomed Hardware Cursor
Zoom Front Buffer
Zoom Back Buffer
60
3D Drawing Engine
CRTC2 and Drawing Engine toggles between 2 buffers for double buffering
21
22
23
24
11
CRTC1
12
CRTC2
31
32
70
Primary Display (Full image display)
Selected Area
71
Secondary Display (Zoomed image)

Figure 4

Figure 5


50
60
3D Drawing Engine
Main Display Buffer
Main Hardware Cursor
....Empty....
Zoom Front Buffer
Zoom Back Buffer
21
22
23
24
11
CRTC1
12
CRTC2
31
32
70
Primary Display (Full image display)
Selected Area
71
Secondary Display (Zoomed image)
Blit the cursor directly onto zoom buffer display

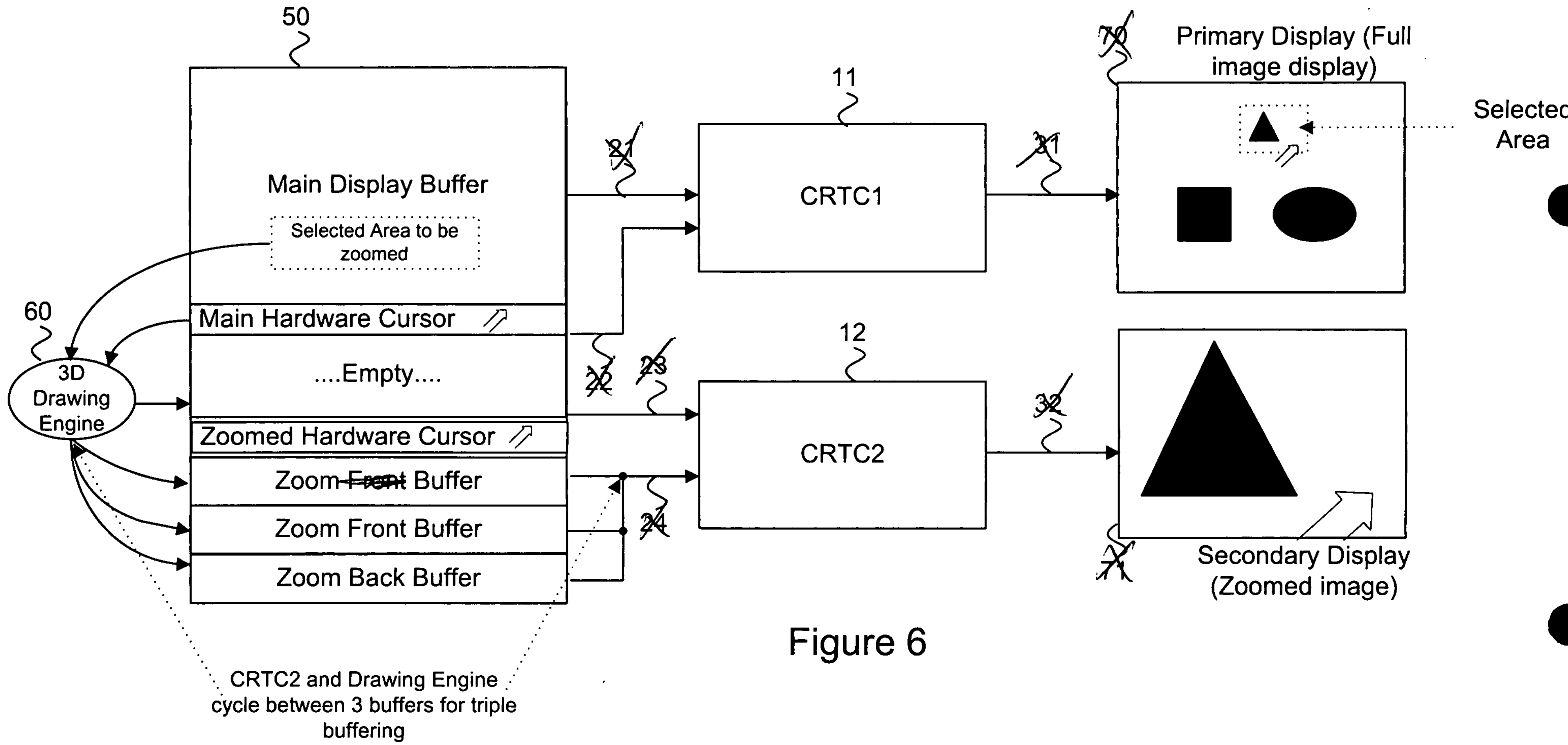

50
Main Display Buffer
Selected Area to be zoomed
Main Hardware Cursor
....Empty....
Zoomed Hardware Cursor
Zoom ~~Front~~ Buffer
Zoom Front Buffer
Zoom Back Buffer
60
3D Drawing Engine
CRTC2 and Drawing Engine cycle between 3 buffers for triple buffering
21
22
23
24
11
CRTC1
12
CRTC2
31
32
70
Primary Display (Full image display)
Selected Area
71
Secondary Display (Zoomed image)

Figure 6

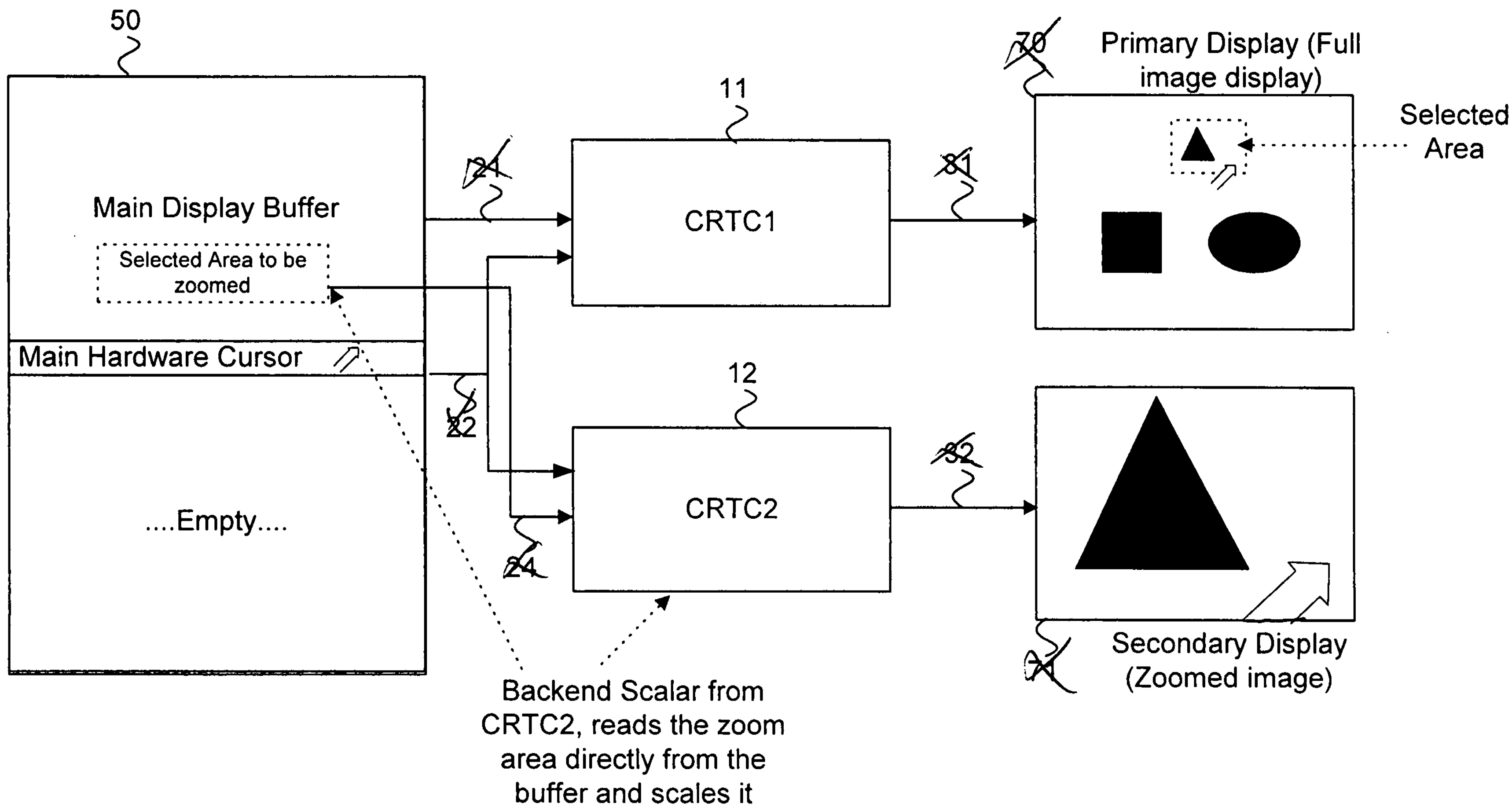

50
Main Display Buffer
Selected Area to be zoomed
Main Hardware Cursor
....Empty....
21
22
24
11
CRTC1
12
CRTC2
81
82
70
Primary Display (Full image display)
Selected Area
Secondary Display (Zoomed image)
71
Backend Scalar from CRTC2, reads the zoom area directly from the buffer and scales it

Figure 7

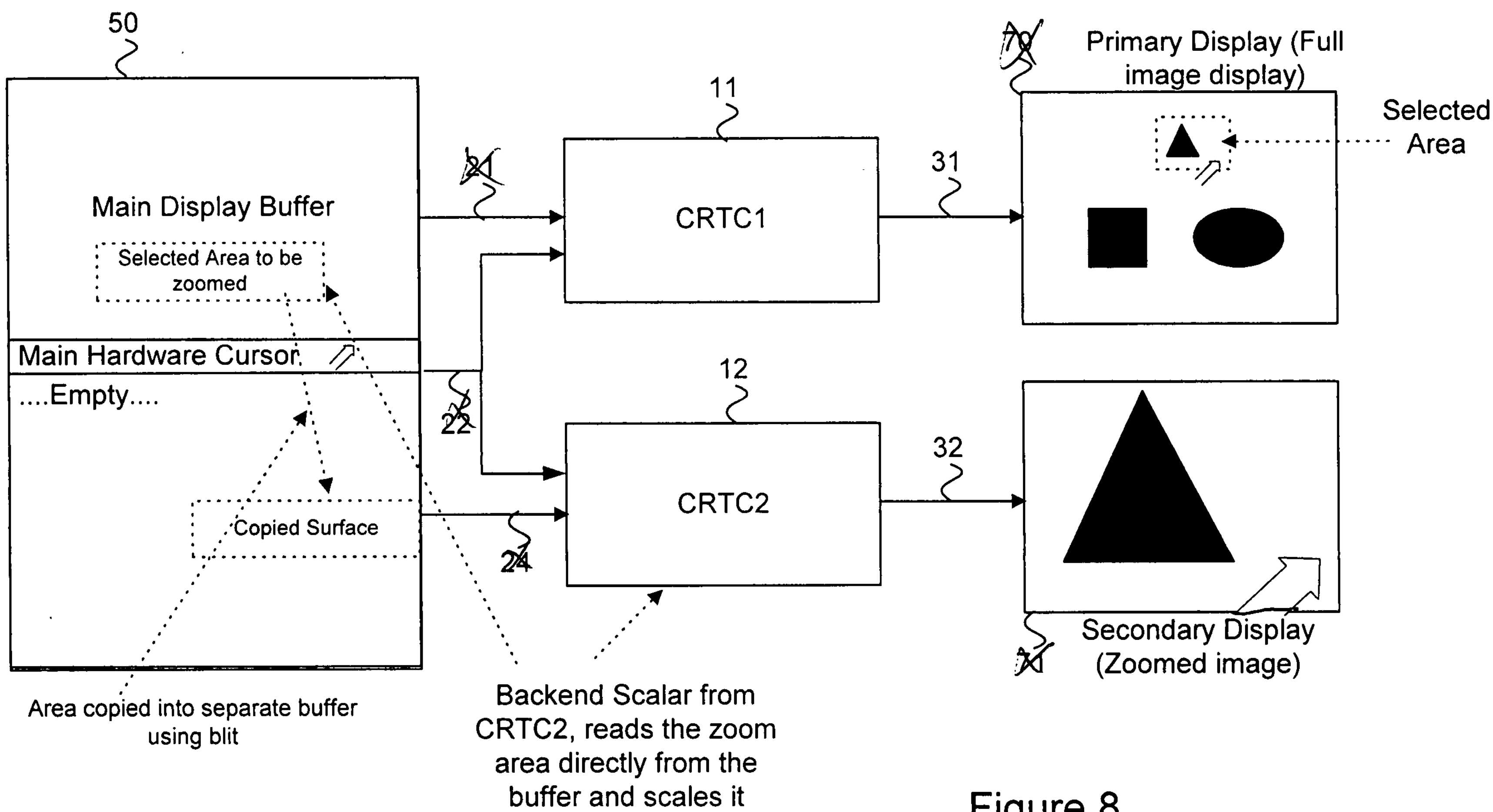

50
Main Display Buffer
Selected Area to be zoomed
Main Hardware Cursor
....Empty....
Copied Surface
Area copied into separate buffer using blit
21
22
24
11
CRTC1
12
CRTC2
Backend Scalar from CRTC2, reads the zoom area directly from the buffer and scales it
31
32
70
Primary Display (Full image display)
Selected Area
71
Secondary Display (Zoomed image)

Figure 8